

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-005791

(43)Date of publication of application : 12.01.2001

(51)Int.Cl.

G06F 15/16

G06F 13/00

(21)Application number : 11-173375

(71)Applicant : NEC CORP

(22)Date of filing : 18.06.1999

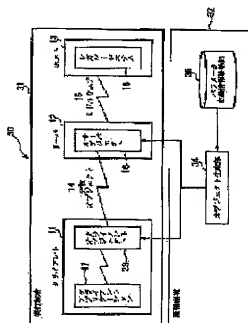
(72)Inventor : OE SHINJI

(54) DISTRIBUTED OBJECT GENERATION SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To generate a distributed object without damaging flexibility to an application program on a client side.

SOLUTION: A client 11 is provided with a client object 33 in the execution environment 31 of a distributed computer environment 30. A client application 17 accesses a server object 16 through a distributed object 14 by using the object 33 and accesses the legacy system 18 of a host 13 by using middleware 15. A development environment 32 consists of an object generating part 34 and a parameter definition information storing part 35. The part 34 generates the objects 33 and a server object 16 on the basis of parameter definition information stored in the part 35 and outputs the objects 33 and 16 to the environment 31.



LEGAL STATUS

[Date of request for examination] 25.05.2000

[Date of sending the examiner's decision of rejection] 17.09.2002

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Data of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(51) Int. Cl. ⁷	識別記号	P I	ページ数 (参考)
G 0 6 F 15/16 13/00	6 2 0 3 5 3	G 0 6 F 15/16 13/00	6 2 0 S 5 B 0 4 5 3 5 3 C 5 B 0 8 9

審査請求 有 請求項の数 2 O L (全 5 頁)

(21) 出願番号 特願平11-173375

(22) 出願日 平成11年6月18日 (1999. 6. 18)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 大江 新次

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100108578

弁理士 高橋 昭男 (外3名)

Fターム (参考) 5B045 G001

5B039 GA01 GA11 GA21 GB03 GB10

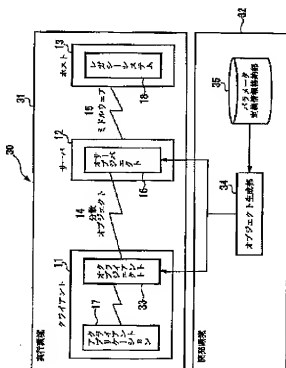
JA11 KA10 KB02 KB04

(54) 【発明の名称】 分散オブジェクト生成方式

(57) 【要約】

【課題】 クライアント側のアプリケーションプログラムに対する汎用性を損なうことなく分散オブジェクトを生成する。

【解決手段】 分散コンピュータ環境30の実行環境31において、クライアント11にクライアントオブジェクト33を備えた。クライアントアプリケーション17は、クライアントオブジェクト33を利用して分散オブジェクト14を介してサーバオブジェクト16を呼び出し、ミドルウェア15を利用してホスト13のレガシーシステム18にアクセスする。開発環境32は、オブジェクト生成部34と、パラメータ定義情報格納部35とから構成した。オブジェクト生成部34はパラメータ定義情報格納部35に格納されたパラメータ定義情報に基づいてクライアントオブジェクト33及びサーバオブジェクト16を生成して実行環境31に出力する。



【特許請求の範囲】

【請求項1】 ミドルウェアを介してホストに結合されるサーバ、分散オブジェクトによりクライアントが結合される分散コンピュータシステムにおいて、

前記クライアントは、前記ホスト上の既存システムに接続する際に使用されるクライアントアプリケーションと、前記分散オブジェクトとのインタフェースを司るクライアントオブジェクトを備えており、

前記分散オブジェクトは、前記クライアントオブジェクトと、前記サーバに備えられたサーバオブジェクトとを結合するように生成されることを特徴とする分散オブジェクト生成方式。

【請求項2】 前記クライアントは、前記クライアントオブジェクトのソースプログラムを生成し、このソースプログラムをコンパイルおよびリンクして前記クライアントオブジェクトを生成するオブジェクト生成手段を備えていることを特徴とする分散オブジェクト生成方式。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、分散コンピュータ環境において、例えばイントラネット内のシステムと、ホストコンピュータ上の既存システム（レガシーシステム）とを結合する分散オブジェクトの生成方式に関する。

【0002】

【従来の技術】従来、分散コンピュータ環境におけるコンネクタ機能は、イントラネット内のシステムとホストコンピュータ上の既存システムを結合する分散オブジェクトの生成機能を有している。この分散オブジェクトを利用する方法としては以下の2つの方法が知られている。

まず、第1の方法について説明する。図9は従来技術の一例による分散コンピュータ環境10を示すブロック構成図である。分散コンピュータ環境10は、クライアント11と、サーバ12と、ホスト13とから構成されている。ここで、クライアント11とサーバ12の間には分散オブジェクト14が設けられており、サーバ12とホスト13の間にはミドルウェア15が設けられている。サーバ12にはサーバオブジェクト16が備えられており、クライアント11のクライアントアプリケーション17は、分散オブジェクト14を介してサーバオブジェクト16を呼び出し、さらに、ミドルウェア15を利用してホスト13のレガシーシステム18にアクセスする。

【0003】次に、第2の方法について説明する。図4は従来技術の一例による分散コンピュータ環境20を示すブロック構成図である。なお、上述した従来技術と同一部分には同じ符号を配して説明を簡略または省略する。ここで、クライアント11には、クライアントアプリケーション17の記述言語に依らずにラッピングモジュール19が設けられており、クライアントアプリケー

ション17は、ラッピングモジュール19を利用することによってサーバオブジェクト16を呼び出す。すなわち、このラッピングモジュール19のオブジェクト呼び出し機能により分散オブジェクト14を介してサーバオブジェクト16を呼び出し、さらに、ミドルウェア15を利用してホスト13のレガシーシステム18にアクセスする。

【0004】

【発明が解決しようとする課題】上記の第1の方法では、クライアント11とサーバ12との間における分散オブジェクト14の例えば通信性能等の性能の改善および変更等を行う際に、分散オブジェクト14のインタフェースの変更が必要となるため、分散オブジェクト14の呼び出し側のクライアントアプリケーション17にも変更が必要となり、クライアントアプリケーション17とのインタフェースに影響を与えてしまうという問題がある。上記の第2の方法では、ラッピングモジュール19はクライアントアプリケーション17の記述言語に依存しており、例えば他言語で記述されたアプリケーションプログラムから利用する場合には、記述言語毎にラッピングモジュール19が必要となり、汎用性に欠けるといふ問題がある。本発明は、上記事情に鑑みてなされたもので、イントラネット内のシステムと既存システムとを結合するコンネクタ機能において、クライアントアプリケーションに対する分散オブジェクトの汎用性を損なうことなく分散オブジェクトを生成することが可能な分散オブジェクト生成方式を提供することを目的とする。

【0005】

【課題を解決するための手段】上記課題を解決して係る目的を達成するために、請求項1に記載の本発明の分散オブジェクト生成方式は、ミドルウェアを介してホストに結合されるサーバに分散オブジェクトによりクライアントが結合される分散コンピュータシステムにおいて、前記クライアントは、前記ホスト上の既存システムに接続する際に使用されるクライアントアプリケーションと、前記分散オブジェクトとのインタフェースを司るクライアントオブジェクトを備えており、前記分散オブジェクトは、前記クライアントオブジェクトと、前記サーバに備えられたサーバオブジェクトとを結合するように生成されることを特徴としている。

【0006】さらに、請求項2に記載の分散オブジェクト生成方式は、前記クライアントは、前記クライアントオブジェクトのソースプログラムを生成し、このソースプログラムをコンパイルおよびリンクして前記クライアントオブジェクトを生成するオブジェクト生成手段を備えていることを特徴としている。

【0007】

【発明の実施の形態】以下、本発明の分散オブジェクト生成方式の実施形態について添付図面を参照しながら説明する。図1は本発明の実施形態に係る分散オブ

3

エクト生成方式を具現する分散コンピュータ環境30のブロック構成図であり、図2は図1に示すオブジェクト生成部12のブロック構成図である。なお、上述した従来技術と同一部分には同じ符号を配して説明を省略または省略する。本実施の形態による分散コンピュータ環境30は、図1に示すように、実行環境31と、開発環境32とから構成されている。実行環境31は、例えばイントラネット内のシステムから既存システムを利用するものであって、クライアント11と、サーバ12と、ホスト13とから構成されている。

【0008】図1に基づいてコネクタ機能を説明すると、クライアント11及びサーバ12はイントラネット内のシステムであり、ホスト13はイントラネット外のホストコンピュータ上のシステムであり、サーバ12はホスト13とミドルウェア15を介して結合されている。ここで、クライアント11と、サーバ12とが分散オブジェクト14で結合されていることによって、クライアント11と、ホスト13のレガシーシステム18とが結合されている。クライアント11はクライアントオブジェクト33を備えており、クライアントアプリケーション17は、クライアントオブジェクト33を利用することによって、分散オブジェクト14を介してサーバオブジェクト16を呼び出す。そして、サーバオブジェクト16はミドルウェア15を利用してホスト13のレガシーシステム18にアクセスする。

【0009】開発環境32は、オブジェクト生成部34と、パラメータ定義情報格納部35とから構成されている。オブジェクト生成部34はパラメータ定義情報格納部35に格納されたパラメータ定義情報に基づいてクライアントオブジェクト33及びサーバオブジェクト16を生成して実行環境31に出力する。図2に示すように、オブジェクト生成部34は、サーバオブジェクトソース生成部41と、クライアントオブジェクトソース生成部42と、サーバオブジェクトソースコンパイル・リンク部43と、クライアントオブジェクトソースコンパイル・リンク部44とから構成されている。

【0010】本実施の形態による分散オブジェクト生成方式を具現する分散コンピュータ環境30は上述の構成を備えており、次に、分散オブジェクト生成方式の動作について図1および図2を参照しながら説明する。先ず、オブジェクト生成部34は、パラメータ定義情報格納部35に格納されたレガシーシステム10に対するパラメータ定義情報に基づいて各オブジェクト16、33を生成する。サーバオブジェクトソース生成部41はパラメータ定義情報に基づいてサーバオブジェクト16のソースプログラムを作成し、次に、サーバオブジェクトソースコンパイル・リンク部43がソースプログラムからサーバオブジェクト16を生成する。さらに、クライアントオブジェクトソース生成部42はパラメータ定義情報に基づいてクライアントオブジェクト33のソース

4

プログラムを作成し、次に、クライアントオブジェクトソースコンパイル・リンク部44がソースプログラムからクライアントオブジェクト33を生成する。次に、開発環境32は、作成した各オブジェクト16、33を実行環境31に出力する。

【0011】実行環境31は、開発環境32から出力されたクライアントオブジェクト33に基づいてクライアントアプリケーション17を作成し、例えばこのクライアントアプリケーション17によりクライアントオブジェクト33及びサーバオブジェクト16を介してホスト13上のレガシーシステム18にアクセスする等を行うことによって、クライアントオブジェクト33及びサーバオブジェクト16の通信性能等に対する評価を行う。ここで、通信性能等の性能に問題があれば、再び、サーバオブジェクトソース生成部41およびクライアントオブジェクトソース生成部42のそれぞれにおいて、サーバオブジェクト16およびクライアントオブジェクト33の各ソースプログラムを修正し、サーバオブジェクトソースコンパイル・リンク部43においてサーバオブジェクト16を、クライアントオブジェクトソースコンパイル・リンク部44においてクライアントオブジェクト33をそれぞれ生成する。なお、クライアントアプリケーション21については、例えば分散オブジェクト14の改善や変更等の影響が及ばないよう作成する。これによって、クライアントアプリケーション21の作成は一度だけで済ますことができる。また、作成した各オブジェクト16、33を、例えば各オブジェクト16、33のソースプログラムの記述言語と異なる他の記述言語によるアプリケーションプログラムから呼び出す場合、アプリケーションプログラムの記述言語がオブジェクト対応すなわち適宜のオブジェクトを組み込んでアプリケーションプログラムを実行することができるものであれば、各オブジェクト16、33を修正すること無しに呼び出し可能となる。

【0012】本実施の形態による分散オブジェクト生成方式によれば、クライアント11にクライアントオブジェクト33が設けられており、例えばクライアント11とサーバ12との間の分散オブジェクト14に改善や変更等を行った場合であっても、クライアントオブジェクト33のみを変更することで対応することができ、クライアントアプリケーション17とのインタフェースに影響を与えることを防ぐことが可能となる。さらに、クライアントオブジェクト33は、オブジェクト対応している記述言語であれば呼び出し可能とされているため、クライアントアプリケーション17がクライアントオブジェクト33と同じ記述言語である必要はなく、クライアントアプリケーション17の記述言語に対する依存度が、例えばラッピングモジュール等に比べて小さくなり、より汎用性を高めることができる。

【0013】

10

20

30

40

50

5

【発明の効果】以上説明したように、本発明の分散オブジェクト生成方式によれば、クライアントにクライアントオブジェクトが設けられており、例えばクライアントとサーバとの間の分散オブジェクトに改修や変更等を行った場合であっても、クライアントオブジェクトのみを変更することで対応することができ、クライアントアプリケーションとのインタフェースに影響を与えることを防ぐことが可能となる。さらに、クライアントオブジェクトは、オブジェクト対応している記述言語であれば呼び出し可能なため、クライアントアプリケーションがクライアントオブジェクトと同じ記述言語である必要はなく、クライアントアプリケーションの記述言語に対する依存度が、例えばラッピングモジュール等に比べて小さくなり、より汎用性を高めることができる。

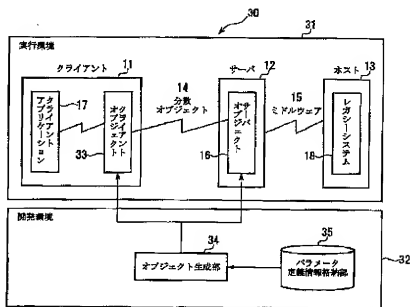
【図面の簡単な説明】

【図1】 本発明の実施形態に係る分散オブジェクト生成方式を具現する分散コンピュータ環境のブロック構成図である。

【図2】 図1に示すオブジェクト生成部のブロック構成図である。

20

【図1】



6

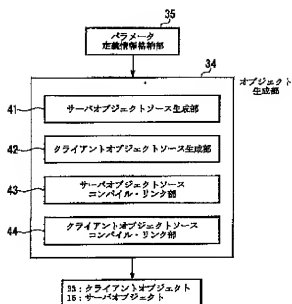
【図3】 従来技術の一例による分散コンピュータ環境を示すブロック構成図である。

【図4】 従来技術の一例による分散コンピュータ環境を示すブロック構成図である。

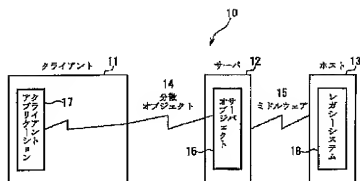
【符号の説明】

- 11 クライアント
- 12 サーバ
- 13 ホスト
- 14 分散オブジェクト
- 15 サーバオブジェクト
- 16 クライアントアプリケーション
- 17 クライアントオブジェクト
- 18 オブジェクト生成部
- 34 パラメータ定義情報格納部
- 35 サーバオブジェクトソース生成部
- 36 クライアントオブジェクトソース生成部
- 37 クライアントオブジェクトソースコンパイル・リンク部
- 38 サーバオブジェクトソースコンパイル・リンク部

【 図2 】



【 図3 】



【 図4 】

